Generalizing Plans to New Environments in Relational MDPs

Thomas Dickerson, Nakul Gopalan, David Meierfrankenfeld

December 12, 2013

Generalizing plans

- learn to plan in one domain
- use this learned knowledge in the next domain using generalizations
- generalizations can be in the form of class based value functions
- this needs the classes to have the same dynamics across domains
- example: Freecraft!

BROWN

Freecraft

- state of an object is defined by its health $(0 - 4)$
- each footman can attack any one of its enemies
- an enemy can attack only its chosen footman
- attacking an object reduces its health
- with n footmen and n enemies the size of the state space is 5^{2n}
- the action space has size n^n

Solving MDPs using linear programs

- a general MDP linear program (LP)

$$\text{Minimize:} \quad \sum_i \alpha(s_i) V(i)$$

$$\text{Subject to:} \quad V(i) \geq R(s_i, a) + \gamma \sum_k P(s'_k | s_i, a) V_k$$

- it has an exponential number of variables 5^{2n} and constraints $5^{2n} n^n$ for our problem
- objective function is often written as a linear combination of basis functions
- there are still exponential number of constraints

Factored MDPs

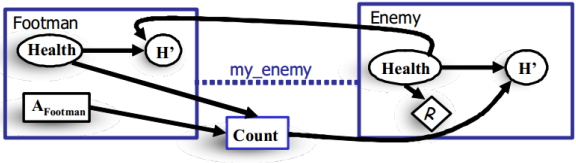
- in some problems the reward and value functions can be written as a sum of subfunctions
- these subfunctions are dependent on a small number of state variables
- in this case an LP can be constructed with the number of constraints exponential in the induced width of the dependency graph[1]
- this reduces the size of the LP dramatically

BROWN

[1] **Efficient Solution Algorithms for Factored MDPs, Guestrin et al.**.

Relational MDPs

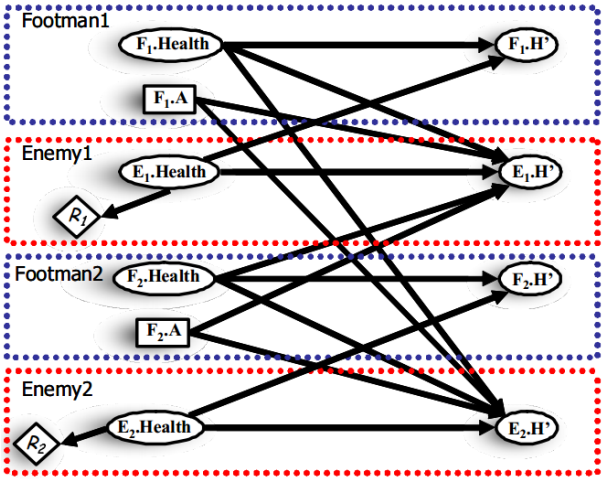
- RMDP defines how state and reward of an object depends on the states and actions of objects linked to it.
- provides class based transition dynamics
- this class structure can be used to define value functions
- class based value functions can be summed up as approximate value functions



Behaviour in tactical domain[2]

[2] **Generalizing Plans to New Environments in Relational MDPs , Guestrin et al..**

Corresponding factored MDPs



BROWN

Factored MDPs and Value functions

- value function for tactical domain

$$\mathbf{V}_{2\text{vs}2}(F1.H, E1.H, F2.H, E2.H) = V_{F1}(F1.H, E1.H) + V_{E1}(E1.H) + V_{F2}(F2.H, E2.H) + V_{E2}(E2.H)$$

- general global value function

$$\mathbf{V}(s) = \sum_{C \in \mathbf{C}} \sum_{o \in O[C]} V_C(S[T_o]) \ ,$$

where o is an object of class C and T_o are its links

LP to solve for Class based value functions

$$\text{Variables:} \quad V_C(t_C) : C \in \mathrm{C}, t_C \in \mathrm{Dom}[\mathbf{T}_C]$$

$$\text{Minimize:} \quad \sum_{C \in \mathrm{C}} \sum_{o \in O[C]} \sum_{t_o \in T_o} P(t_o) V_C(t_o)$$

$$\text{Subject to:} \quad \sum_{C \in \mathrm{C}} \sum_{o \in O[C]} V_C(s[T_o]) \geq \sum_{o \in O[C]} R^C(s[S_o], a[o.A]) + \gamma \sum_{s'} P(s'|s,a) \sum C \in \mathrm{C} \sum_{o \in O[C]} V_C(s'[T_o])$$

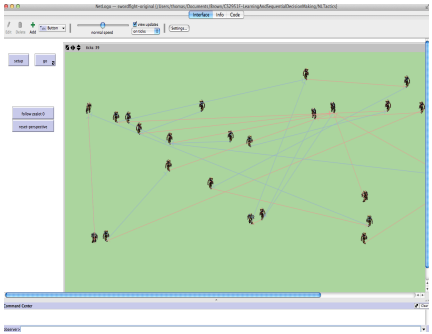
Implementation

- implemented the model domain on Netlogo and Burlap
- implemented joint action, class based value and reward framework in Burlap
- implemented a class based vlaue function solver using LP in Burlap

BROWN

the only plots we have

Original Plot



Our Plot

BROWN

Not really

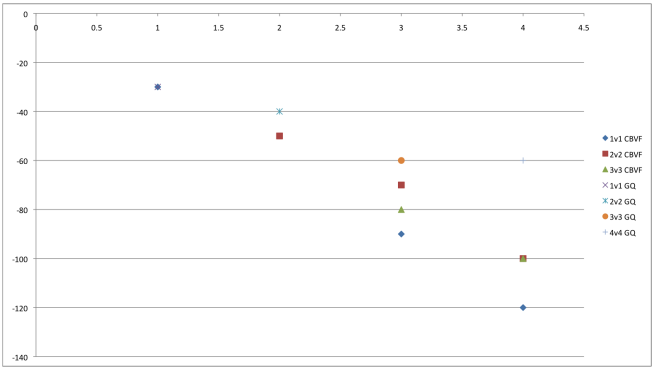


Figure: players vs total rewards

BROWN

Memory

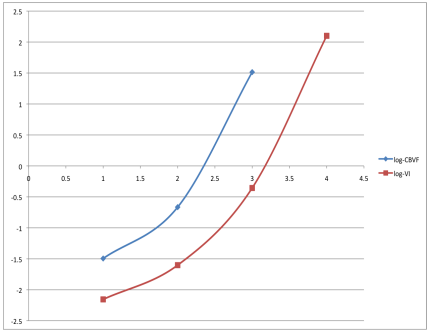
```
// RUN THIS CODE FOR A GOOD TIME

System.out.println("");

zdg = new ZealotsDomainGenerator(4,0,4);
Domain fourVFour = zdg.generateDomain();
start = System.currentTimeMillis();
zdg.getFactoredSolution(fourVFour, 0.95);
```

Process Name	Memory ▼	Threads	Ports
java	8.12 GB	24	213
kernel_task	1.10 GB	99	0
Firefox	991.7 MB	46	251
Eclipse	764.8 MB	121	1,708

Memory usage on a fancy apple computer!

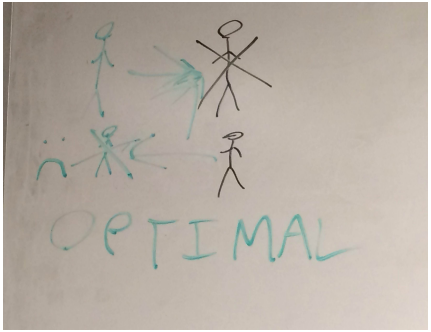


Runtime Plot log time vs # agents per side

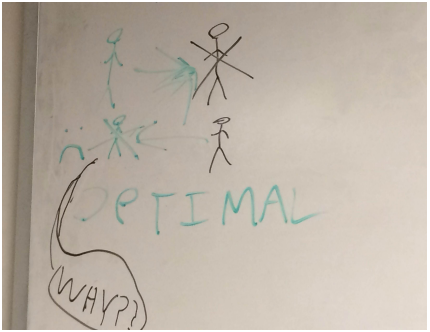
BROWN

Pain slides!

```
keys.contains(keys.iterator().next()) -> false
```



Why??



Beating the dead horse

BROWN

- The Good
  1. class based value functions to approximate MDP as an idea
  2. solving LPs with exponential constraints
- The Bad
  1. tactical domains are oversimplified to favour the Footman class
  2. the paper claims the values do not generalize well to larger set of objects
- The Ugly
  1. lack of clarity in the paper about the LP solving process
  2. no software released